

Deep Learning Classification of Hotels to Combat Human Trafficking

Project Category: Computer Vision

Jacob Smith
jacobas@stanford.edu

Justin Lim
jlim23@stanford.edu

Ryan Beauchamp
rmb87@stanford.edu

Abstract

The problem of locating human trafficking victims using photos taken in a hotel room with thousands of potential locations is high stakes and an intractable problem for the human eye. In addition to the challenge of identifying a single hotel from an image of the room, there is the added difficulty of the victim partially obfuscating the image. We built on previous work by comparing the performance of multiclass classification using a pretrained ResNet-34 model with a contrastive learning approach using triplet loss. Overall, ResNet-34 had the highest top-5 accuracy at 69.4% compared to 46.8% using triplet loss. Nonetheless, contrastive learning is still a promising approach for further exploration, as it scales to new hotels without retraining the model.

1 Introduction

Human trafficking is a horrid reality for over 40.3 million people worldwide [1]. Often, photos of victims will be posted from hotel rooms, but the sheer volume of possible hotels makes it the equivalent of searching for a needle in a haystack. The goal of this project was to develop a deep learning model to identify the hotel location of the victim based on an image, thereby enabling law enforcement to take action. The input to the model was an image of a hotel room with obfuscation added to imitate having a victim in the photograph. We used two neural network architectures to output the top-5 most likely hotel locations. First, we used contrastive learning with a triplet loss function to generate image embeddings and picked the 5 classes with the smallest L2 embedding distances. Second, we used a multiclass classification model with a Softmax classifier to generate a probability per class and selected the 5 classes with the largest probabilities.

2 Related work

2.1 Contrastive Learning with Image Embeddings

Prior work [2][3] suggests that using pretrained convolutional neural networks (CNNs) to convert images into vector embeddings, and then selecting the most similar embedding from a database of hotel image embeddings, will be effective for predicting hotel-IDs from an image. This is an approach that has been successfully applied in other fields, most notably facial verification and recognition [4]. The benefit of this approach is it excels at one-shot learning [5]. However, previous applications [3] note that it is challenging to "enforce a large margin between hotel chains." We used 128-dimensional image embeddings (same as [3]) as part of a contrastive learning model with a triplet loss function (see 4 below).

2.2 Multiclass Classification with ResNet

Others [6][7] have classified hotel images for a variety of tasks using a pretrained ResNet model with a Softmax classifier. The benefit of this approach is it is more computationally efficient to train relative to contrastive learning with triplet loss (see 4 below). However, since there are over three thousand classes with an average of only fourteen images per class (and as few as two images per class), training the model to accurately predict a probability for each class presents a challenge. We used a pretrained ResNet-34 model with two fully connected layers added at the end of the model for our multiclass classification task (see 4 below).

3 Dataset and Features

Our data was sourced from our project’s corresponding Kaggle competition [9]. There are 44,692 unmasked images of the 3,105 different hotels linked to their respective hotel ID’s, which we split 80% into the training set and 20% into the validation set stratified over the classes.

Our training and validation sets were stratified by class to make sure that the proportions of examples for each hotel-ID were preserved. We resized our images to be 448 x 252 pixels, then randomly flipped the images across the horizontal axis with a probability of 0.5. For each image, we randomly applied the 4,950 different training masks on top of the training and validation images, imitating the presence of victims in the photographs. We then normalized the training and validation sets based on the mean and standard deviation of the training set.

4 Methods

To establish a baseline for our task, we used contrastive learning with triplet loss. We used a ResNet-34 pretrained model and froze the first 6 layers to preserve basic image understanding and speed up training. We added a final fully connected layer from 512 to 128 neurons to reduce the output to a 128-dimensional embedding which can be compared to a database of hotel image embeddings precomputed by the model. We trained the model with triplet loss, passing into the model triplets composed of anchor image a , positive image from the same class p , and negative image from a different class n (see Figure 1). The triplet loss function is shown in (1), with d representing the L2 norm between the image embeddings and the margin representing a tunable hyperparameter of the model, with 1.0 being the default.

$$L(a, p, n) = \max\left(d(a, p) - d(a, n) + \text{margin}, 0\right) \quad (1)$$

This loss function encourages larger inter-class distances and smaller intra-class distances. Thus, we would expect images of the same hotel location to have closer embeddings than images from different hotels. A top-5 prediction is made by choosing the 5 classes in the database with the closest L2 norm distance.

For our multiclass classification approach, we used a ResNet-34 pretrained model with the first 6 layers frozen. We then added two fully-connected layers to increase the number of neurons, first from 512 to 1,024 neurons, and then from 1,024 to 3,105 neurons, with each neuron representing a hotel class. A dropout probability of 0.5 was added to the final fully connected layer before the output layer to help combat overfitting (see 5 below). We used cross entropy loss which computes the loss using a log Softmax output over the number of classes. A top-5 prediction is made by choosing the 5 classes in the database with the highest Softmax probabilities (see Figure 2).

5 Experiments/Results/Discussion

Top-5 accuracy was the primary metric used to evaluate the models since the dataset was relatively balanced between classes, with the largest class representing 3.1% of the dataset and only three additional classes representing more than 1% of the dataset. Evaluating using the top-5 accuracy is a tradeoff between limiting the number of hotel locations for law enforcement to investigate while minimizing false negatives. For each model’s baseline, we used a batch size of 32, a stochastic gradient descent optimizer with a 0.05 base learning rate α , a learning rate decay γ of 0.1, and a step size of 10 epochs.



Figure 1: For contrastive learning with triplet loss, a hotel image representing the anchor image, positive image from the same class, and negative image from a different class are passed into a ResNet-34 pretrained model with a fully connected layer added to produce 128-dimensional image embeddings.

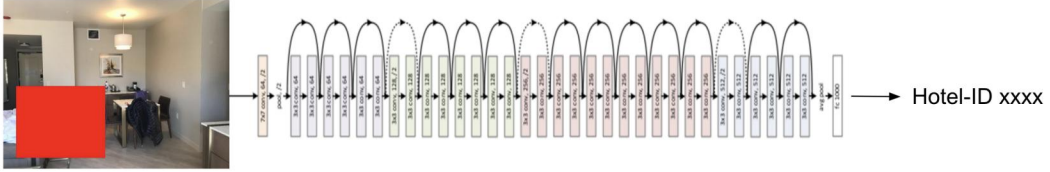


Figure 2: For multiclass classification, a hotel image is passed into the ResNet-34 neural network, which features special skip connections and utilizes batch normalization [9]. Two fully connected layers were added at the end of the model with dropout to combat overfitting. The model provides one output for each class, with the largest output determining classification by hotel-ID.

As a baseline, our contrastive learning model achieved a 97.7% top-5 training accuracy and a 46.8% top-5 validation accuracy. This indicates overfitting to the training set, which could be improved with regularization techniques such as dropout and weight decay. However, this model was computationally expensive to train, taking ~ 3 days to complete 20 epochs. With more computational resources, we could train longer and tune hyperparameters to achieve higher top-5 accuracy.

For our multiclass classification model, as an early estimate of the upper bound of performance we trained the model without obfuscation and were able to achieve 99.6% top-5 training accuracy and 71.2% top-5 validation accuracy. After adding obfuscations, we were able to maintain nearly the same performance, with 98.9% top-5 training accuracy and a 69.4% top-5 validation accuracy. This proved that it was possible to successfully identify a hotel location with a partially obfuscated image of the room. In addition, this model was not only more effective than the contrastive learning model at our key success metric, but also more computationally efficient to train, taking ~ 1 day to complete 20 epochs.

Similar to contrastive learning, our multiclass classification model also overfit to the training set. To help combat this, we experimented with regularization techniques including adding dropout to the final fully connected layer before the output layer and including weight decay. We also explored



Figure 3: For the validation image on the left, the model correctly predicted the class that includes the image on the right. The similarity of the bed spread, lamps, and wall color can help explain how the model was able to correctly predict this hotel-ID.



Figure 4: For the validation image on the left, the model incorrectly predicted the class that includes the image on the right. The glare on the two hotel’s wall art and the limited information included can help explain this incorrect prediction.

ways to guide the model to find a better minimum in the loss function such as a one cycle learning rate scheduler, which ramped the model from a learning rate near 0.005 up to a maximum learning rate of 0.05, and then slowly stepped down over time. The goal was to allow stochastic gradient descent to tune with a small learning rate before ramping up to give the model flexibility to find a better local minimum, and then to decrease the learning rate to allow gradient descent to settle at that minimum [10]. Overall, dropout and weight decay each slowed convergence, extending the required training time. With 17 epochs and a dropout probability of 0.5, we achieved 83.3% top-5 training accuracy and 63.6% top-5 validation accuracy, demonstrating that this approach successfully reduced the gap between training and validation accuracy. With additional training time, this approach is on track to remove the observed variance and improve the top-5 validation accuracy.

Our multiclass classification model also achieved a top-1 accuracy of 47.13% with a top-1 precision, recall, and F1-score of 28.66%, 30.61%, and 27.80%, respectively. As a result, the model is able to correctly predict the hotel-ID as its first choice nearly half of the time. In addition, 383 classes have perfect top-1 precision and 381 classes have perfect top-1 recall in the model. At the same time, about half of the hotels are in the 0-10% range for each of the top-1 metrics (see Figure 5). Given the consequences of a false negative for the task at hand, this was a compelling reason for using top-5 accuracy as the success metric.

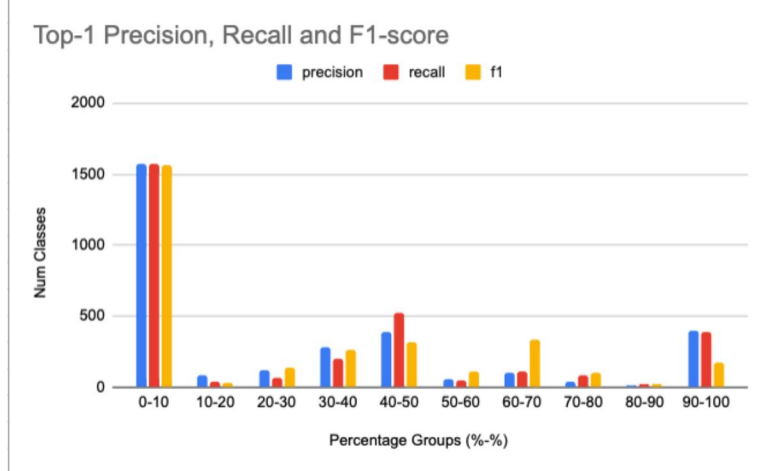


Figure 5: Counts of top-1 precision, recall, and F1-score for individual classes grouped by percentages. The model’s top-1 accuracy of 47.13% explains just over half of the classes being in the 0-10% range. Given the task difficulty and high number of classes, top-5 accuracy was used as the success metric.

From reviewing examples the model classified, when there is sufficient visual clues about the hotel location, the model was often able to correctly make a prediction, such as in Figure 3 where the bed spread, lamps, and wall colors match. In contrast, when presented with limited information or superficial similarities, such as the glare on wall art in Figure 4, the model was unable to make a correct prediction. This incorrect prediction may also have occurred because the predicted class has many examples of wall art in the training set. Conducting data augmentation to have a more even split of wall art between classes and preprocessing the data to remove visual noise, such as glare, could aid the model’s predictive capabilities.

6 Conclusion/Future Work

Identifying hotel locations from images partially obfuscated by victims is a critical task for law enforcement in combating human trafficking. We have shown that deep learning can be a useful tool for identifying such hotel locations through both contrastive learning and multiclass classification, with the latter achieving a top-5 validation accuracy of 69.4%. We find these results to be promising, particularly given the significant time required to identify a hotel location manually.

Multiclass classification outperformed contrastive learning for a few reasons. First, it has the advantage of combining information across multiple images from a class, whereas contrastive learning compares individual images within a class. As a result, when comparing individual image embeddings, there might be a more similar individual image from a different class even if in aggregate the images from the same class share more in common. This issue arises due to high intra-class variability (such as between the bedroom and the bathroom) combined with inter-class similarities (see wall art example in Figure 4), which could be alleviated with more images per class in the database. In addition, we were only able to run the contrastive learning model once for 20 epochs due to the significant computational resources required (see 3 above).

With additional computational resources, the contrastive learning model could be trained for more epochs and a more thorough exploration of the hyperparameter space could be conducted to achieve even better performance. First, we’d aim to explore regularization techniques to close the variance gap between training and validation performance. Next, we’d explore alternative approaches to triplet loss that appear promising from the literature, including a Siamese Network architecture and Additive Angular Margin Loss (ArcFace) [3]. While it takes much longer to train, contrastive learning is a promising approach given its natural scalability to new hotels that it did not encounter during training time. This advantage also makes contrastive learning a more useful tool for law enforcement in combating human trafficking around the world.

7 Acknowledgements

Much thanks to Professors Ng and Katanforoosh, TAs Yanjun Chen and Tianhe Yu, and the rest of the CS 230 staff for their support throughout the quarter. We would also like to thank Kaggle for hosting the competition and sharing the dataset.

8 Contributions

Jacob

Conducted literature review of approaches similar to our baseline using larger datasets. Created our HotelImagesDataset class which our Dataloader uses to load the training and validation images. Created an index used by the dataset because the size of our training set cannot be efficiently loaded into memory otherwise. Helped with programming the multi-class classification approach, specifically implemented our accuracy calculation section using a top-5 accuracy metric and improved model logging. Ran baseline models and collected statistics. Added triplet loss as well as the contrastive learning model. Created the associated Datasets and Dataloaders for triplet loss. Worked on metrics calculations and outputting examples that were correctly/incorrectly predicted.

Justin

Made our HotelTrainDataloader class to create our dataloader to be used in training. This includes creating the transformations, calculating the mean and standard deviation for normalization, and adding random obfuscations onto all images. Split the data into a training and validation set with stratification. Set up the argument parser for the main file. Helped set up connection to the AWS instance. Set up test function for ResNet including metric calculations and outputting examples that were correctly/incorrectly predicted. Helped tune learning rate, learning rate decay, and step size.

Ryan

Conducted literature review to determine baseline model and subsequent approaches to explore. Added logistic regression and SVM models for initial testing. Programmed software architecture for Resnet34 multiclass classification model. Helped implement top-5 accuracy as the success metric. Programmed software architecture for contrastive learning with triplet loss. Helped implement the associated datasets and dataloaders. Helped implement error analysis functions. Set up AWS instance with GPU support to run the models with GPU acceleration. Ran models on AWS for contrastive learning with triplet loss and multiclass classification. Tuned hyperparameters including normalization, dropout, weight decay, optimizer, scheduler, learning rate, learning rate decay, and step size.

Everyone contributed equally to this paper.

References

- [1] International Labour Organization. "Global estimates of modern slavery: Forced labour and forced marriage." International Labour Organization (2017).
- [2] Jain, Prachi. "Combat Human Trafficking by similar Hotels Recognition using images." Becoming Human: Artificial Intelligence Magazine, August 2021, <https://becominghuman.ai/combat-human-trafficking-by-similar-hotels-recognition-using-images-e5e67e66c060>.
- [3] Tseytlin, Boris, and Ilya Makarov. "Hotel Recognition via Latent Image Embeddings." International Work-Conference on Artificial Neural Networks. Springer, Cham, 2021.
- [4] Taigman, Yaniv, et al. "Deepface: Closing the gap to human-level performance in face verification." Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.
- [5] "Face Verification/Recognition, One Shot Learning (Online Learning)." Hamhochi, <https://dothanhblog.wordpress.com/2020/02/26/face-verification-one-shot-learning-online-learning/>.
- [6] Dylski, Kornel. "Transfer learning in practice. Image classification for hotel images with fast.ai library." nexocode, February 21, 2019, <https://nexocode.com/blog/posts/transfer-learning-in-practice/>.
- [7] Hotel Image Recognition and Classification Solution. Nexocode, <https://nexocode.com/case-studies/hotel-image-recognition-and-classification-solution/>.

[8] "Hotel-ID to Combat Human Trafficking 2022-FGVC9." Kaggle, <https://www.kaggle.com/competitions/hotel-id-to-combat-human-trafficking-2022-fgvc9>.

[9] CS231n Convolutional Neural Networks for Visual Recognition, <https://cs231n.github.io/convolutional-networks/>.

[10] Finding Good Learning Rate and The One Cycle Policy, <https://towardsdatascience.com/finding-good-learning-rate-and-the-one-cycle-policy-7159fe1db5d6>.